

Risk Minimization Framework for Multiple Instance Learning from Positive and Unlabeled Bags

Han Bao¹, Tomoya Sakai^{2,3}, Issei Sato^{2,1,3}, and Masashi Sugiyama^{3,2,1}

¹ Department of Computer Science, The University of Tokyo

² Department of Complexity Science and Engineering, The University of Tokyo

³ Center for Advanced Intelligence Project, RIKEN

{tsutsumi@ms., sakai@ms., sato@sugi@}k.u-tokyo.ac.jp

Abstract. Multiple instance learning (MIL) is a variation of traditional supervised learning problems where data (referred to as bags) are composed of sub-elements (referred to as instances) and only bag labels are available. MIL has a variety of applications such as content-based image retrieval, text categorization and medical diagnosis. Most of the previous work for MIL assume that the training bags are fully labeled. However, it is often difficult to obtain an enough number of labeled bags in practical situations, while many unlabeled bags are available. A learning framework called PU learning (positive and unlabeled learning) can address this problem. In this paper, we propose a convex PU learning method to solve an MIL problem. We experimentally show that the proposed method achieves better performance with significantly lower computational costs than an existing method for PU-MIL.

1 Introduction

Multiple instance learning (MIL) [8] is a learning problem with *bags* and *instances*. Instances are the same as the traditional feature vectors, while bags are sets of instances. The numbers of instances in bags vary among the bags. Bag labels are defined as follows:

- If a given bag contains *at least one positive instance*, then its label is positive.
- If a given bag contains *no positive instances*, then its label is negative.

This is the basic setup of MIL. The goal of MIL is to predict labels of test bags. Here instance labels are unavailable, which makes MIL more difficult than the ordinary classification setting. MIL is applicable to a wide range of real-world problems such as molecule behavior prediction [3,15], drug activity prediction [8], domain theory [7], content-based image retrieval [17,13,25], visual tracking [2], object detection [18,12], text categorization [1] and medical diagnosis [9,22].

A lot of approaches for MIL have been developed [8,16,26,1,11,23], which are classified into two groups in general:

- (1) The methods in the first group are based on generative modeling, including

the diverse density [16] and its extension, the expectation-maximization diverse density (EM-DD) [26]. These methods find out an instance close to training positive bags and far from training negative bags, which is referred to as the concept point. This process is carried out by gradient-based search from every training instance, which is computationally inefficient.

(2) The methods in the second group are based on discriminative modeling. The multiple-instance support vector machine (MI-SVM) [1] is an approach based on SVMs. Empirical evaluation shows that MI-SVM performs well on MIL datasets, but its optimization problem is non-convex and computationally inefficient. The key-instance support vector machine (KI-SVM) [14] converts MI-SVM into a convex problem by using a discrete vector, but it is still difficult to optimize due to a large number of optimization variables. Gärtner et al. [11] introduced set kernels (a.k.a. multiple instance kernels), which are extensions of the standard kernel functions to MIL. The set kernels can be used to construct a standard SVM classifier, which performs well on MIL datasets. The optimization problem in this training procedure is convex and the global solution can be obtained efficiently.

In this work, we propose a novel method to construct the multiple instance classifiers only from positive and unlabeled bags, while the above standard approaches to MIL assume that training bags are fully labeled. This problem is called PU-MIL. For example, PU-MIL is considered to be applicable to the following situations:

- The situation where it is difficult to obtain an enough amount of labeled data due to the significant labeling cost, such as outlier detection. In outlier detection based on supervised classification, it is often difficult to label all outlier samples, so we can label a few outlier samples as positive and regard the other samples as unlabeled.
- The situation where the true negative labels are essentially unavailable, such as music album recommendation, where we try to predict a music album that attracts a consumer from their purchase history. Each album contains some songs, which can be regarded as multiple instances. Whether the album is attracting the consumer or not corresponds to positive and negative, but only whether the consumer bought (positive) or not (unlabeled) is available⁴.

Our contribution in this paper is to propose a novel PU-MIL method based on the risk minimization framework [20]. Together with a linear-in-parameter model, the proposed method is formulated as a convex optimization problem, and the global optimal solution can be computed efficiently. To the best of our knowledge, this is the first convex PU-MIL method (see Table 1). Through experiments, we demonstrate that the proposed method combined with the minimax kernel [1] compares favorably with an existing method.

The rest of this paper is structured as follows. In Sect. 2, we review existing methods for PU learning [19,20] and MIL [11], on which our proposed method

⁴ The fact that the consumer did not buy an album does not directly mean that the album is not attracting (negative). The consumer might not buy just because he or she did not notice it.

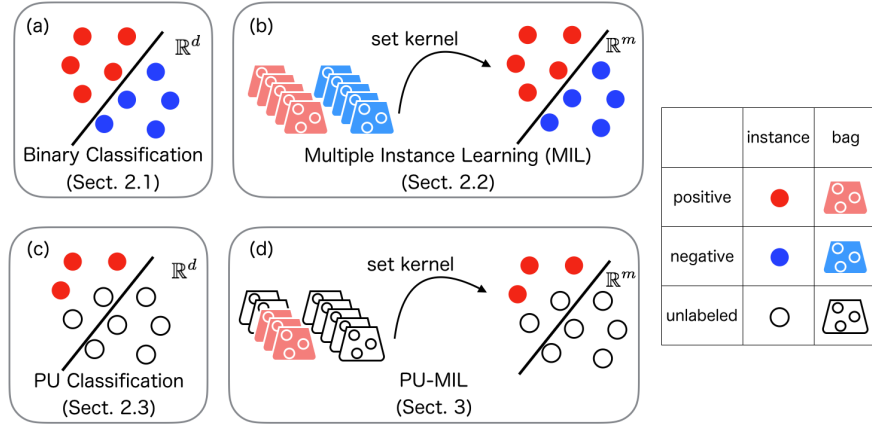
Table 1. Comparison of existing and proposed discriminative methods for MIL.

Learning from \ Convexity	Convex	Non-convex
Positive and Negative	set kernels [11] sMIL [4] KI-SVM [14] miGraph [27]	MI-SVM [1] MissSVM [28] soft-bag SVM [13] dMIL [25]
Positive and Unlabeled	PU-SKC (Sect. 3)	puMIL [24]

is based. In Sect. 3, we explain the formulation and optimization algorithm of our proposed method, called the *positive and unlabeled set kernel classifier* (PU-SKC). In Sect. 4, we experimentally compare the proposed method (PU-SKC) with the existing method (puMIL) [24]. Finally, we conclude this work in Sect. 5.

2 Problem Formulation and Related Work

In this section, we formulate the problems we discuss in this paper (see Fig. 1) and review related works.

Fig. 1. A schematic of problems. In this work, we consider (d) multiple instance learning from positive and unlabeled bags (PU-MIL).

2.1 Ordinary Binary Classification

First, we define some notations. Let $\mathbf{x} \in \mathbb{R}^d$ be a d -dimensional feature vector and $y \in \{+1, -1\}$ be its corresponding class label. In the ordinary binary classification problem, we construct a binary classifier

$$\begin{aligned} f(\mathbf{x}) &= \text{sign}(g(\mathbf{x})) \in \{+1, -1\}, \\ g : \mathbb{R}^d &\rightarrow \mathbb{R}, \end{aligned} \quad (1)$$

from a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where N is the number of training samples. Here we use a linear-in-parameter model for g :

$$g(\mathbf{x}) = \boldsymbol{\alpha}^\top \boldsymbol{\phi}(\mathbf{x}) + \beta,$$

where \top denotes the transpose, $\boldsymbol{\alpha} \in \mathbb{R}^m$ is an m -dimensional parameter vector, $\beta \in \mathbb{R}$ is a bias parameter and $\boldsymbol{\phi} : \mathbb{R}^d \rightarrow \mathbb{R}^m$ is a vector of basis functions. The support vector machine (SVM) [6] is one of the standard methods for training a binary classifier. The optimization problem of SVM is as follows:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^m, \beta \in \mathbb{R}} \quad \frac{1}{2} \|\boldsymbol{\alpha}\|^2 + C \sum_{i=1}^N \max \{0, 1 - y_i (\boldsymbol{\alpha}^\top \boldsymbol{\phi}(\mathbf{x}_i) + \beta)\}, \quad (2)$$

where $C > 0$ is a penalty parameter. This problem can be reformulated as the quadratic programming, which can be solved efficiently.

2.2 Multiple Instance Learning

Here we formulate the problem of multiple instance learning (MIL) and review an existing method.

Formulation Hereafter $\mathcal{P}(A)$ denotes the power set⁵ of A . Let $X = \{\mathbf{x}_i | \mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^n \in \mathcal{P}(\mathbb{R}^d) \setminus \emptyset$ be a bag containing n instances whose dimensions are d , and $Y \in \{+1, -1\}$ be a bag label corresponding to X . The problem is to construct a binary classifier:

$$\begin{aligned} f(X) &= \text{sign}(g(X)) \in \{+1, -1\}, \\ g : \mathcal{P}(\mathbb{R}^d) \setminus \emptyset &\rightarrow \mathbb{R}, \end{aligned} \quad (3)$$

from a fully-labeled training dataset $\mathcal{D} = \{(X_b, Y_b)\}_{b=1}^N$, where N denotes the number of bags in \mathcal{D} .

⁵ The power set of A is a set of all subsets of A , including \emptyset and A itself. In the MIL setting, bags belong to $\mathcal{P}(\mathbb{R}^d)$, i.e., bags are composed of some elements in \mathbb{R}^d .

Multiple Instance Kernels Gärtner et al. [11] proposed set kernels (multiple instance kernels). These kernels map bags (sets of instances) to feature spaces. A type of the set kernels called the statistic kernel \tilde{k} is defined as follows:

$$\tilde{k}(X, X') := k(\mathbf{s}(X), \mathbf{s}(X')),$$

where k is an arbitrary kernel function such as the Gaussian kernel, and \mathbf{s} is called the statistics. For example, the following minimax statistics is a typical choice:

$$\mathbf{s}_{\min\text{-max}}(X) := \left[\min_{\mathbf{x} \in X} x^{(1)}, \dots, \min_{\mathbf{x} \in X} x^{(d)}, \max_{\mathbf{x} \in X} x^{(1)}, \dots, \max_{\mathbf{x} \in X} x^{(d)} \right]^\top, \quad (4)$$

where $x^{(i)}$ is the i -th element of an instance \mathbf{x} in the bag X . Gärtner et al. [11] showed that the statistic kernel with the minimax statistics (4) for \mathbf{s} and the polynomial kernel for k outperforms other set kernels in the standard MIL:

$$\tilde{k}_{\min\text{-max}}(X, X') := (\mathbf{s}_{\min\text{-max}}(X)^\top \mathbf{s}_{\min\text{-max}}(X') + 1)^\rho, \quad (5)$$

where ρ is a positive integer. The statistic kernel (5) is referred to as the minimax kernel. We can then construct the following set kernel classifier g :

$$g(X) = \boldsymbol{\alpha}^\top \boldsymbol{\phi}(X) + \beta, \quad (6)$$

$$\boldsymbol{\phi}(X) = \begin{bmatrix} \tilde{k}_{\min\text{-max}}(X, C_1) \\ \vdots \\ \tilde{k}_{\min\text{-max}}(X, C_M) \end{bmatrix},$$

where C_1, \dots, C_M are kernel centers and M is the number of kernel centers. The standard MIL can be solved by using the classifier (6) with SVM (2).

2.3 Learning from Positive and Unlabeled Data

Here we formulate the binary classification problem of learning from positive and unlabeled instances and review existing methods.

Formulation We assume positive samples $\{\mathbf{x}_i^P\}_{i=1}^{N_P}$ and unlabeled samples $\{\mathbf{x}_i^U\}_{i=1}^{N_U}$ are generated as follows:

$$\begin{aligned} \{\mathbf{x}_i^P\}_{i=1}^{N_P} &\stackrel{\text{i.i.d.}}{\sim} p_+(\mathbf{x}) := p(\mathbf{x}|y = +1), \\ \{\mathbf{x}_i^U\}_{i=1}^{N_U} &\stackrel{\text{i.i.d.}}{\sim} p(\mathbf{x}) := \pi p(\mathbf{x}|y = +1) + (1 - \pi)p(\mathbf{x}|y = -1) \\ &= \pi p_+(\mathbf{x}) + (1 - \pi)p_-(\mathbf{x}), \end{aligned} \quad (7)$$

where $\pi := p(y = +1)$ is called the class prior. Our objective is to construct a binary classifier (1) only from positive samples and unlabeled samples.

Learning Instance-Level Classifiers From Positive and Unlabeled Data

du Plessis et al. [19,20] proposed methods based on the empirical risk minimization framework to learn only from positive and unlabeled samples. In the ordinary binary classification setting, the Bayes optimal classifier g^* minimizes the following misclassification rate:

$$R_{0-1}(g) = \pi \mathbb{E}_P [\ell_{0-1}(g(\mathbf{x}))] + (1 - \pi) \mathbb{E}_N [\ell_{0-1}(-g(\mathbf{x}))], \quad (8)$$

where $\mathbb{E}_P [\cdot]$ and $\mathbb{E}_N [\cdot]$ denote the expectations over $p_+(\mathbf{x})$ and $p_-(\mathbf{x})$ respectively and ℓ_{0-1} denotes the zero-one loss:

$$\ell_{0-1}(z) = \begin{cases} 0 & \text{if } z \geq 0, \\ 1 & \text{otherwise.} \end{cases}$$

In practice, the misclassification rate (8) is difficult to optimize since the sub-gradient of ℓ_{0-1} is always 0 except at $z = 0$. For this reason, we usually use a *surrogate* loss function⁶. Then the risk function R with the surrogate loss function ℓ is written as

$$R(g) := \pi \mathbb{E}_P [\ell(g(\mathbf{x}))] + (1 - \pi) \mathbb{E}_N [\ell(-g(\mathbf{x}))]. \quad (9)$$

Since negative samples are not available in the PU learning setup, let us consider expressing the risk (9) without $\mathbb{E}_N [\cdot]$. By the definition of the unlabeled sample distribution (7), the following holds:

$$(1 - \pi) \mathbb{E}_N [\ell(-g(\mathbf{x}))] = \mathbb{E}_U [\ell(-g(\mathbf{x}))] - \pi \mathbb{E}_P [\ell(-g(\mathbf{x}))].$$

Substituting this into the risk (9), we obtain

$$R(g) = \pi \mathbb{E}_P [\ell(g(\mathbf{x})) - \ell(-g(\mathbf{x}))] + \mathbb{E}_U [\ell(-g(\mathbf{x}))]. \quad (10)$$

If the surrogate loss function ℓ satisfies

$$\ell(z) - \ell(-z) = -z, \quad (11)$$

the risk (10) can be written as

$$R(g) = \pi \mathbb{E}_P [-g(\mathbf{x})] + \mathbb{E}_U [\ell(-g(\mathbf{x}))]. \quad (12)$$

The risk (12) is convex if the surrogate loss function ℓ is convex. Convex loss functions such as the squared loss ℓ_S , the logistic loss ℓ_{LL} and the double hinge loss ℓ_{DH} satisfy condition (11):

$$\begin{aligned} \ell_S(z) &= \frac{1}{4}(z - 1)^2 - \frac{1}{4}, \\ \ell_{LL}(z) &= \log(1 + \exp(-z)), \\ \ell_{DH}(z) &= \max\left(-z, \max\left(0, \frac{1 - z}{2}\right)\right). \end{aligned} \quad (13)$$

We use the risk (12) to obtain a convex formulation of PU-MIL in Sect. 3.

⁶ For example, the hinge loss $\ell_H(z) = \max(-z, 0)$ and the ramp loss $\ell_{RH}(z) = \frac{1}{2} \max(0, \min(2, 1 - z))$ are commonly used.

3 Positive and Unlabeled Set Kernel Classifier

In this section, we propose a convex method for PU-MIL, named the PU-SKC (positive and unlabeled set kernel classifier).

3.1 Multiple Instance Learning from Positive and Unlabeled Bags

Here we formulate the problem of multiple instance learning from positive and unlabeled bags (PU-MIL).

The purpose of PU-MIL is to construct the bag-level classifier (3) from a positively labeled training dataset $\mathcal{D}_P = \{(X_b^P, Y_b = +1)\}_{b=1}^{N_P}$ and an unlabeled training dataset $\mathcal{D}_U = \{X_{b'}^U\}_{b'=1}^{N_U}$, where N_P and N_U denotes the number of positive bags in \mathcal{D}_P and the number of unlabeled bags in \mathcal{D}_U , respectively.

3.2 Formulation

As we mentioned in Sect. 2.3, du Plessis et al. [19,20] formulated the PU learning problem in the empirical risk minimization framework. If we use a loss function $l(z)$ such that $l(z) - l(-z) = -z$, we have the following objective function:

$$J(g) = \pi \mathbb{E}_P [-g(X)] + \mathbb{E}_U [\ell(-g(X))]. \quad (14)$$

Here we use a linear-in-parameter model with the set kernel function as a classifier:

$$g(X) = \boldsymbol{\alpha}^\top \boldsymbol{\phi}(X) + \beta, \quad (15)$$

where $\boldsymbol{\phi}$ is the vector of basis functions:

$$\boldsymbol{\phi}(X) = \begin{bmatrix} \tilde{k}_{\min-\max}(X, X_1^P) \\ \vdots \\ \tilde{k}_{\min-\max}(X, X_{N_P}^P) \\ \tilde{k}_{\min-\max}(X, X_1^U) \\ \vdots \\ \tilde{k}_{\min-\max}(X, X_{N_U}^U) \end{bmatrix}. \quad (16)$$

$\tilde{k}_{\min-\max}$ is the polynomial minimax kernel (5). Similarly to the standard binary classification, we predict a given bag as positive if $g(X) \geq 0$, and as negative if $g(X) < 0$.

The risk (14) together with the bag-level classifier (15) and the l_2 regularizer induces the following objective function to be minimized by the sample averages:

$$\begin{aligned} \hat{J}(\boldsymbol{\alpha}, b) = & \pi \cdot \frac{1}{N_P} \sum_{b=1}^{N_P} (-\boldsymbol{\alpha}^\top \boldsymbol{\phi}(X_b^P) - \beta) \\ & + \frac{1}{N_U} \sum_{b'=1}^{N_U} \ell_{\text{DH}}(-\boldsymbol{\alpha}^\top \boldsymbol{\phi}(X_{b'}^U) - \beta) + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \boldsymbol{\alpha}, \end{aligned} \quad (17)$$

where $\lambda \geq 0$ is the regularization parameter. Here we use the double hinge loss ℓ_{DH} (13) since du Plessis et al. [20] reported that it achieved the best performance in the ordinary PU classification setting. Note that π is the bag-level class prior, i.e., $\pi = p(Y = +1)$, which must be estimated from the training data. We explain how to estimate it in Sect. 3.3. This problem can be rewritten in the form of quadratic programming by using slack variables $\boldsymbol{\xi}$ as

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \beta, \boldsymbol{\xi}} \quad & -\frac{\pi}{N_P} \mathbf{1}^\top \Phi_P \boldsymbol{\alpha} - \pi \beta + \frac{1}{N_U} \mathbf{1}^\top \boldsymbol{\xi} + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{\xi} \geq \mathbf{0}, \\ & \boldsymbol{\xi} \geq \frac{1}{2} \mathbf{1} + \frac{1}{2} \Phi_U \boldsymbol{\alpha} + \frac{1}{2} \beta \mathbf{1}, \\ & \boldsymbol{\xi} \geq \Phi_U \boldsymbol{\alpha} + \beta \mathbf{1}, \end{aligned} \quad (18)$$

where \geq for vectors denotes the element-wise inequality and $\mathbf{0}, \mathbf{1}$ denote the all-zero and all-one vector, respectively. Matrices Φ_P and Φ_U are defined as follows:

$$\Phi_P = \begin{bmatrix} \boldsymbol{\phi}^\top(X_1^P) \\ \vdots \\ \boldsymbol{\phi}^\top(X_{N_P}^P) \end{bmatrix}, \quad \Phi_U = \begin{bmatrix} \boldsymbol{\phi}^\top(X_1^U) \\ \vdots \\ \boldsymbol{\phi}^\top(X_{N_U}^U) \end{bmatrix}.$$

3.3 Bag-Level Class Prior Estimation

A bag-level class prior estimation algorithm can be obtained by a simple extension of the instance-level version explained in Appendix A. The difference is basis functions used in the empirical estimator of $r(\mathbf{x})$. We use the polynomial minimax kernel (5) to obtain the bag-level basis functions (16). Then the bag-level class prior π can be estimated similarly.

4 Experiments

In this section, we experimentally compare the proposed method⁷ with the existing method (see Appendix B) and give answers to the following research questions:

- Q1: Does the proposed method perform well regardless of the true class prior?
Q2: Is the proposed method computationally efficient?

4.1 Dataset

We use standard MIL datasets: Musk and Corel⁸. The details about these benchmark datasets are shown in Table 2.

⁷ Implementation is published at <https://github.com/levelfour/pumil>.

⁸ <http://www.cs.columbia.edu/~andrews/mil/datasets.html>

Table 2. Details of datasets: The last two rows indicate the numbers of instances per bag, which are the average with the standard deviation.

Number of	Musk1	Musk2	Elephant	Fox	Tiger
features	166	166	230	230	230
positive bags	47	39	100	100	100
negative bags	45	63	100	100	100
positive instances	2.3 (2.6)	10.0 (26.1)	3.8 (4.2)	3.2 (3.6)	2.7 (3.1)
negative instances	2.9 (6.9)	54.7 (176.0)	3.2 (3.6)	3.4 (3.8)	3.4 (3.8)

Algorithm 1 Generation of Training / Test Sets for Benchmark MIL Datasets

Input: \mathcal{D}_P : original positive bags, \mathcal{D}_N : original negative bags, π : true bag-level class prior, L : $\#\{\text{labeled positive bags}\}$, U : $\#\{\text{unlabeled bags}\}$, T : $\#\{\text{test bags}\}$

- 1: $\mathcal{D}_L \subset \mathcal{D}_P$ $\triangleright |\mathcal{D}_L| = L$
- 2: $\mathcal{D}_P := \mathcal{D}_P \setminus \mathcal{D}_L$
- 3: $N_U^P \sim B(U + T, \pi)$
- 4: $N_U^N := U + T - N_U^P$
- 5: $\mathcal{D}_U := \mathcal{D}_P'(\subset \mathcal{D}_P) \cup \mathcal{D}_N'(\subset \mathcal{D}_N)$ $\triangleright |\mathcal{D}_P'| = N_U^P, |\mathcal{D}_N'| = N_U^N$
- 6: $\mathcal{D}_U' \subset \mathcal{D}_U$ $\triangleright |\mathcal{D}_U'| = T$
- 7: $\mathcal{D}_U := \mathcal{D}_U \setminus \mathcal{D}_U'$

Output: $\mathcal{D}_L \cup \mathcal{D}_U$: training set, \mathcal{D}_U' : test set

Since these datasets are too small to evaluate PU methods, we augment them to increase the number of bags. Specifically, bags chosen randomly from the original datasets are duplicated and then Gaussian noise with mean zero and variance 0.01 is added to each dimension. In this way, we increase the number of samples in Musk (Musk1 and Musk2) 10 times and Corel (Elephant, Fox and Tiger) 5 times. After this augmentation process, we generate a training set (including labeled positive bags and unlabeled bags) and a test set. This generation process is described in Algorithm 1 (we set $L = 20, U = 180, T = 200$).

4.2 Evaluation Method

Hyperparameters (the degree parameter ρ in the polynomial kernel (5) and the regularization parameter λ in the objective function (17)) are selected via 5-fold cross-validation from $\rho \in [1, 2, 3]$ and $\lambda \in [10^0, 10^{-3}, 10^{-6}]$. We use the area under the receiver operating characteristics curve (AUC) [5] as the evaluation metric, which is robust to the class imbalance, i.e., the situation where the class prior π takes extremely high or low values. AUC is defined as follows:

$$\begin{aligned}
 \text{AUC} &:= \mathbb{E}_P [\mathbb{E}_N [I(g(\mathbf{x}^P) \geq g(\mathbf{x}^N))]] \\
 &= 1 - \mathbb{E}_P [\mathbb{E}_N [I(g(\mathbf{x}^P) < g(\mathbf{x}^N))]] \\
 &= 1 - \mathbb{E}_P [\mathbb{E}_N [\ell_{0-1}(g(\mathbf{x}^P) - g(\mathbf{x}^N))]] ,
 \end{aligned}$$

where $I(\cdot)$ denotes the indicator function. We define the following risk (called the AUC risk) to be minimized:

$$R_{\text{AUC}}(g) = \mathbb{E}_{\mathbf{P}} [\mathbb{E}_{\mathbf{N}} [\ell_{0-1}(g(\mathbf{x}^{\mathbf{P}}) - g(\mathbf{x}^{\mathbf{N}}))]] .$$

Here we have to calculate the AUC risk only from the positive and unlabeled data. We have the following relation from the definition of the marginal probability:

$$\begin{aligned} \mathbb{E}_{\mathbf{P}} [\mathbb{E}_{\mathbf{U}} [\ell_{0-1}(g(\mathbf{x}^{\mathbf{P}}) - g(\mathbf{x}^{\mathbf{U}}))]] &= \pi \mathbb{E}_{\mathbf{P}} [\mathbb{E}_{\mathbf{P}'} [\ell_{0-1}(g(\mathbf{x}^{\mathbf{P}}) - g(\mathbf{x}^{\mathbf{P}'}))]] \\ &\quad + (1 - \pi) \mathbb{E}_{\mathbf{P}} [\mathbb{E}_{\mathbf{N}} [\ell_{0-1}(g(\mathbf{x}^{\mathbf{P}}) - g(\mathbf{x}^{\mathbf{N}}))]] , \end{aligned}$$

where $\mathbb{E}_{\mathbf{P}} [\cdot] = \mathbb{E}_{\mathbf{x}^{\mathbf{P}} \sim p_+(\mathbf{x})} [\cdot]$ and $\mathbb{E}_{\mathbf{P}'} [\cdot] = \mathbb{E}_{\mathbf{x}^{\mathbf{P}'} \sim p_+(\mathbf{x})} [\cdot]$. Then we obtain the following formulation of the AUC risk only from positive and unlabeled data (PU-AUC risk):

$$\begin{aligned} R_{\text{PU-AUC}}(g) &= \frac{1}{1 - \pi} \mathbb{E}_{\mathbf{P}} [\mathbb{E}_{\mathbf{U}} [\ell_{0-1}(g(\mathbf{x}^{\mathbf{P}}) - g(\mathbf{x}^{\mathbf{U}}))]] \\ &\quad - \frac{\pi}{1 - \pi} \mathbb{E}_{\mathbf{P}} [\mathbb{E}_{\mathbf{P}'} [\ell_{0-1}(g(\mathbf{x}^{\mathbf{P}}) - g(\mathbf{x}^{\mathbf{P}'}))]] . \end{aligned}$$

This risk is used for the cross-validation.

4.3 Results

In this section, we show the experimental results and give answers to the research questions.

A1: The proposed method outperforms the existing method regardless of the true bag-level class prior.

Table 3 shows averages with standard deviations of the predictions, which are carried out 50 times under each class prior. Bold faces represent the best methods under each class prior. This is tested by the one-sided t-test with 5% significant level (first the higher method is chosen as the best method, then the other method is checked whether it is competitive or not by the one-sided t-test). As it can be seen from Table 3, PU-SKC outperforms the existing method puMIL [24] under the different class priors. Note that the true class prior in Table 3 means the predefined value for dataset generation (see Algorithm 1), not the estimated class prior during the learning process (see Sect. 3.3).

A2: The proposed method is much more computationally efficient than the existing method.

Next, we compare the execution time between the proposed method and the existing method. The result is shown in Figure 2. This result shows PU-SKC is much more computationally efficient than the existing method.

Table 3. AUC: Each result is the average with the standard deviation over 50 trials. Bold faces represent the best methods under each setting (tested by the one-sided 5% t-test).

dataset	true class prior	PU-SKC	puMIL [24]
Musk1	0.1	0.928 (0.051)	0.839 (0.075)
	0.4	0.860 (0.063)	0.750 (0.086)
	0.7	0.731 (0.100)	0.645 (0.114)
Musk2	0.1	0.911 (0.042)	0.815 (0.076)
	0.4	0.862 (0.047)	0.700 (0.092)
	0.7	0.711 (0.104)	0.673 (0.091)
Elephant	0.1	0.857 (0.057)	0.752 (0.083)
	0.4	0.785 (0.076)	0.690 (0.071)
	0.7	0.685 (0.087)	0.633 (0.096)
Fox	0.1	0.625 (0.095)	0.584 (0.073)
	0.4	0.599 (0.060)	0.571 (0.048)
	0.7	0.529 (0.064)	0.544 (0.057)
Tiger	0.1	0.816 (0.052)	0.736 (0.067)
	0.4	0.766 (0.057)	0.646 (0.066)
	0.7	0.679 (0.094)	0.612 (0.082)

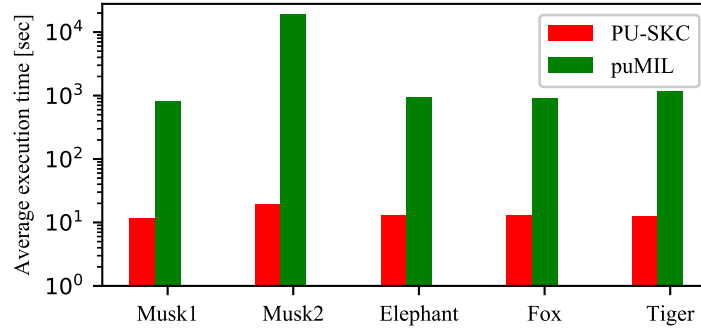
5 Conclusion

Multiple instance learning has versatile applications such as content-based image retrieval, text categorization and medical diagnosis. In this work, we considered a multiple instance learning problem when only positive bags and unlabeled bags are available, which does not require all training bags to be labeled. We proposed a convex method, PU-SKC, to solve PU multiple instance classification. This method is based on the convex formulation of PU learning [20] and the set kernel [11]. PU-SKC performed better than the existing PU multiple instance classification method [24] in the classifications of the benchmark MIL datasets (drug activity prediction and image annotation). Furthermore, we confirmed the proposed method was much more computationally efficient than the existing method through the experiment.

References

1. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: Advances in Neural Information Processing Systems 15. pp. 577–584. MIT Press, Cambridge, MA, USA (2002)
2. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 983–990 (2009)
3. Buchanan, B.G., Mitchell, T.M.: Model-directed learning of production rules. Tech. rep., Stanford, CA, USA (1977)

Fig. 2. Average execution time: each result is the average execution time of 50 trials under the bag-level class prior $\pi = 0.1$. PU-SKC is executed on the given hyperparameter.



4. Bunesco, R.C., Mooney, R.J.: Multiple instance learning for sparse positive bags. In: Proceedings of the 24th International Conference on Machine Learning. pp. 105–112. ACM, New York, NY, USA (2007)
5. Cortes, C., Mohri, M.: AUC optimization vs. error rate minimization. In: Proceedings of the 16th International Conference on Neural Information Processing Systems. pp. 313–320. MIT Press, Cambridge, MA, USA (2003)
6. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* 20(3), 273–297 (1995)
7. Dietterich, T.G., Flann, N.S.: An inductive approach to solving the imperfect theory problem. In: Proceedings of the American Association for Artificial Intelligence Spring Symposium Series: Explanation-Based Learning. pp. 42–46 (1988)
8. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence* 89(1), 31–71 (1997)
9. Dundar, M.M., Fung, G., Krishnapuram, B., Rao, R.B.: Multiple-instance learning algorithms for computer-aided detection. *IEEE Transactions on Biomedical Engineering* 55(3), 1015–1021 (2008)
10. Fu, Z., Robles-Kelly, A., Zhou, J.: MILIS: Multiple instance learning with instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(5), 958–977 (2011)
11. Gärtner, T., Flach, P.A., Kowalczyk, A., Smola, A.J.: Multi-instance kernels. In: Proceedings of the 19th International Conference on Machine Learning. pp. 179–186. Morgan Kaufmann (2002)
12. Kanezaki, A., Harada, T., Kuniyoshi, Y.: Scale and rotation invariant color features for weakly-supervised object learning in 3D space. In: International Conference on Computer Vision Workshops. pp. 617–624 (2011)
13. Li, W., Vasconcelos, N.: Multiple instance learning for soft bags via top instances. In: The IEEE Conference on Computer Vision and Pattern Recognition (2015)
14. Li, Y.F., Kwok, J.T., Tsang, I.W., Zhou, Z.H.: A convex method for locating regions of interest with multi-instance learning. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II. pp. 15–30. ECML PKDD '09 (2009)

15. Lindsay, R.K.: Applications of artificial intelligence for organic chemistry : the DENDRAL project. McGraw-Hill Book Co New York (1980)
16. Maron, O., Lozano-Pérez, T.: A framework for multiple-instance learning. In: Advances in Neural Information Processing Systems 10. pp. 570–576. MIT Press, Cambridge, MA, USA (1998)
17. Maron, O., Ratan, A.L.: Multiple-instance learning for natural scene classification. In: Proceedings of the 15th International Conference on Machine Learning. pp. 341–349. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998)
18. Pandey, M., Lazebnik, S.: Scene recognition and weakly supervised object localization with deformable part-based models. In: Proceedings of the 2011 International Conference on Computer Vision. pp. 1307–1314. IEEE Computer Society, Washington, DC, USA (2011)
19. du Plessis, M.C., Niu, G., Sugiyama, M.: Analysis of learning from positive and unlabeled data. In: Advances in Neural Information Processing Systems 27, pp. 703–711. Curran Associates, Inc. (2014)
20. du Plessis, M.C., Niu, G., Sugiyama, M.: Convex formulation for learning from positive and unlabeled data. In: Proceedings of the 32nd International Conference on Machine Learning. pp. 1386–1394. JMLR Workshop and Conference Proceedings (2015)
21. du Plessis, M.C., Sugiyama, M.: Class prior estimation from positive and unlabeled data. IEICE Transactions on Information and Systems 97(5), 1358–1362 (2014)
22. Tong, T., Wolz, R., Gao, Q., Guerrero, R., Hajnal, J.V., Rueckert, D.: Multiple instance learning for classification of dementia in brain MRI. Medical Image Analysis 18(5), 808–818 (2014)
23. Wang, J., Zucker, J.D.: Solving the multiple-instance problem: A lazy learning approach. In: Proceedings of the 17th International Conference on Machine Learning. pp. 1119–1126. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2000)
24. Wu, J., Zhu, X., Zhang, C., Cai, Z.: Multi-instance learning from positive and unlabeled bags. In: Proceeding of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 237–248. Springer International Publishing (2014)
25. Wu, J., Yinan Yu, Chang Huang, Kai Yu: Deep multiple instance learning for image classification and auto-annotation. IEEE Conference on Computer Vision and Pattern Recognition pp. 3460–3469 (2015)
26. Zhang, Q., Goldman, S.A.: EM-DD: An improved multiple-instance learning technique. In: Advances in Neural Information Processing Systems 13. pp. 1073–1080. MIT Press (2001)
27. Zhou, Z.H., Sun, Y.Y., Li, Y.F.: Multi-instance learning by treating instances as non-i.i.d. samples. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 1249–1256. ACM, New York, NY, USA (2009)
28. Zhou, Z.H., Xu, J.M.: On the relation between multi-instance learning and semi-supervised learning. In: Proceedings of the 24th International Conference on Machine Learning. pp. 1167–1174. ACM, New York, NY, USA (2007)

A Instance-Level Class Prior Estimation

Here the estimation of the class prior $\pi = p(y = +1)$ is addressed.

In the research by du Plessis and Sugiyama [21], a class prior estimation method from positive and unlabeled data by partial matching was proposed. This method estimates the class prior π by minimizing the Pearson (PE) divergence from positive data distribution $\pi p(\mathbf{x}|y = 1)$ to unlabeled data distribution $p(\mathbf{x})$:

$$\begin{aligned}\pi^* &= \arg \min_{\pi} \text{PE}[\pi p(\mathbf{x}|y = 1)|p(\mathbf{x})] \\ &= \arg \min_{\pi} \frac{1}{2} \int \left(\frac{\pi p(\mathbf{x}|y = 1)}{p(\mathbf{x})} - 1 \right)^2 p(\mathbf{x}) d\mathbf{x}.\end{aligned}$$

It is still possible to estimate $p(\mathbf{x}|y = 1)$ from positive samples and $p(\mathbf{x})$ from unlabeled samples using, e.g., the kernel density estimation, but du Plessis and Sugiyama [21] empirically showed that such a naive approach often does not produce a good estimator of π . A better approach is to lower bound the PE divergence and directly maximize it:

$$\begin{aligned}\text{PE}[\pi p(\mathbf{x}|y = 1)|p(\mathbf{x})] &\geq \pi \int r(\mathbf{x}) p(\mathbf{x}|y = 1) d\mathbf{x} \\ &\quad - \frac{1}{2} \int r(\mathbf{x})^2 p(\mathbf{x}) d\mathbf{x} - \pi + \frac{1}{2},\end{aligned}\tag{19}$$

where $r(\mathbf{x})$ is an arbitrary function. In practice, a linear-in-parameter model based on Gaussian kernel basis functions is used to estimate r :

$$\hat{r}(\mathbf{x}) = \boldsymbol{\alpha}^\top \boldsymbol{\phi}(\mathbf{x}).$$

Our goal is to find the tightest lower bound by maximizing the right-hand side of (19) at first with respect to r and then minimize it with respect to π . The former maximization problem with the l_2 regularizer can be written as

$$\begin{aligned}\hat{\boldsymbol{\alpha}} &= \arg \max_{\boldsymbol{\alpha}} \left[\pi \boldsymbol{\alpha}^\top \mathbf{h} - \frac{1}{2} \boldsymbol{\alpha}^\top H \boldsymbol{\alpha} - \pi + \frac{1}{2} - \frac{\lambda}{2} \boldsymbol{\alpha}^\top \boldsymbol{\alpha} \right] \\ &= \arg \max_{\boldsymbol{\alpha}} \left[\pi \boldsymbol{\alpha}^\top \mathbf{h} - \frac{1}{2} \boldsymbol{\alpha}^\top H \boldsymbol{\alpha} - \frac{\lambda}{2} \boldsymbol{\alpha}^\top \boldsymbol{\alpha} \right],\end{aligned}\tag{20}$$

$$\text{where } H = \int \boldsymbol{\phi}(\mathbf{x}) \boldsymbol{\phi}(\mathbf{x})^\top p(\mathbf{x}) d\mathbf{x}, \quad \mathbf{h} = \int \boldsymbol{\phi}(\mathbf{x}) p(\mathbf{x}|y = 1) d\mathbf{x},$$

and $\lambda \geq 0$ denotes the regularization parameter. In practice, H and \mathbf{h} are estimated by the sample averages:

$$\hat{H} = \frac{1}{N_P} \sum_{i=1}^{N_P} \boldsymbol{\phi}(\mathbf{x}_i^P) \boldsymbol{\phi}(\mathbf{x}_i^P)^\top, \quad \hat{\mathbf{h}} = \frac{1}{N_U} \sum_{j=1}^{N_U} \boldsymbol{\phi}(\mathbf{x}_j^U).$$

Using these estimators, Eq. (20) can be reformulated as follows:

$$\hat{\boldsymbol{\alpha}} = \arg \max_{\boldsymbol{\alpha}} \left[\pi \boldsymbol{\alpha}^\top \hat{\mathbf{h}} - \frac{1}{2} \boldsymbol{\alpha}^\top \hat{H} \boldsymbol{\alpha} - \frac{\lambda}{2} \boldsymbol{\alpha}^\top \boldsymbol{\alpha} \right].$$

An analytical solution to the above problem can be obtained as follows:

$$\hat{\alpha} = \pi \hat{G}^{-1} \hat{\mathbf{h}}, \quad \hat{G} = \hat{H} + \lambda I,$$

where I denotes the identity matrix. This leads to the following PE divergence estimator:

$$\widehat{\text{PE}} = \pi^2 \hat{\mathbf{h}}^\top \hat{G}^{-1} \hat{\mathbf{h}} - \pi^2 \frac{1}{2} \hat{\mathbf{h}}^\top \hat{G}^{-1} \hat{H} \hat{G}^{-1} \hat{\mathbf{h}} - \pi + \frac{1}{2}.$$

Then the analytical minimizer of $\widehat{\text{PE}}$ can be obtained as

$$\hat{\pi} = \left[2 \hat{\mathbf{h}}^\top \hat{G}^{-1} \hat{\mathbf{h}} - \hat{\mathbf{h}}^\top \hat{G}^{-1} \hat{H} \hat{G}^{-1} \hat{\mathbf{h}} \right]^{-1}.$$

B Multiple Instance Learning from Positive and Unlabeled Bags

Wu et al. [24] proposed an instance-level method for the PU-MIL problem (called puMIL). As discussed in the previous work [1,14], the key instance (the most positive instance) in each bag is important in MIL. This is also the case in PU-MIL, but the problem is that we cannot tell the labels of unlabeled bags. To address this issue, Wu et al. [24] introduced the bag confidence, which describes how much confident the given unlabeled bag is negative. Let $\nu_b \in [0, 1]$ be the confidence of the b -th bag. Then the learning problem of an SVM classifier $g(X) = \mathbf{w}^\top \tilde{\mathbf{x}}$ ($\tilde{\mathbf{x}} \in X$) under the given $\boldsymbol{\nu}$ is formulated as

$$\min_{\mathbf{w} \in \mathbb{R}^d, \{\tilde{\mathbf{x}}_b\}_{b=1}^{N_P+N_U}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{b=1}^{N_P+N_U} \max(0, 1 - \nu_b \tilde{Y}_b \mathbf{w}^\top \tilde{\mathbf{x}}_b), \quad (21)$$

where \tilde{Y}_b is the estimated bag label ($\tilde{Y}_b = +1$ for a positive bag and $\tilde{Y}_b = -1$ for a possibly negative bag coming from the set of unlabeled bags) and $\tilde{\mathbf{x}}_b$ is the key instance in the bag X_b . ν_b weighs the contribution of $\tilde{\mathbf{x}}_b$ according to its confidence. For positive bags, the confidence is simply set as $\nu_b = 1$. The overall training method is summarized as follows:

1. Initialize L distinct bag confidences $\boldsymbol{\nu}^{(1)}, \dots, \boldsymbol{\nu}^{(L)}$, where L is the number of the bag confidences.
2. Extract N_P *possibly* negative bags from unlabeled bags.
3. Make a positive margin pool (PMP), which consists of the key instances from positive bags and possibly negative bags.
4. Solve (21) with using the PMP to obtain an SVM classifier, which is evaluated by the F-measure.

In Step 2, unlabeled bags with the highest confidences are extracted and regarded as possibly negative. Bag confidences are initialized randomly and updated via a genetic algorithm (see Step 4 and (23)).

In Step 3, first the generative distribution of negative instances is estimated by a weighted kernel density estimator [10]:

$$\hat{p}(\mathbf{x}|y = -1) = \frac{1}{\tilde{N}_N} \sum_{b=1}^{\tilde{N}_N} \sum_{\mathbf{x}' \in \tilde{X}_b^N} k(\mathbf{x}, \nu_b \mathbf{x}'), \quad (22)$$

where \tilde{N}_N is the number of all instances included in the extracted bags in Step 2 and k denotes a kernel function and \tilde{X}_b^N is the b -th possibly negative bag. Using the estimator (22), we can obtain a PMP by extracting the key instances from bags:

$$\tilde{\mathbf{x}}_b = \arg \min_{\mathbf{x}_b \in X_b} \hat{p}(\mathbf{x}_b|y = -1).$$

After building the PMP, an SVM classifier can be obtained in Step 4. We evaluate the given bag confidences ν by calculating the F-measures⁹. Here we have L bag confidences, and we update them by a genetic algorithm. First, the bag confidence with the highest F-measure (denoted by $\hat{\nu}$) is cloned to replace the other confidences under the fixed rate $c \in [0, 1]$. This process, called the mutation, is caused by

$$\mathbf{v}_l = \nu_l + r(1 - f_l)(\hat{\nu} - \nu_l), \quad (23)$$

where r is drawn from the standard normal distribution, ν_l is the l -th bag confidence and f_l is the F-measure obtained from ν_l . ν_l is to be replaced by \mathbf{v}_l if \mathbf{v}_l produces a better result. We iterate Steps 2 – 4 until the best bag confidence $\hat{\nu}$ converges or the number of iterations reaches the predefined limit.

After the above training steps, the best bag confidence $\hat{\nu}$ is to be obtained. We use $\hat{\nu}$ for the initial bag confidence and iterate Steps 2 – 4 again to obtain a classifier for test prediction¹⁰.

However, the process to extract possibly negative bags from unlabeled bags based on the bag confidences (Step 2) is not shown to converge to the optimal solution in the previous work, while this method works well experimentally. In the following section, we use the PU risk minimization framework (12) to formulate PU-MIL.

⁹ True bag labels are needed to calculate the F-measure, but those of unlabeled bags are unavailable. This point was not discussed in Wu et al. [24], so we assume that possibly negative bags have negative labels and the other bags in the unlabeled set have positive labels when we calculate the F-measure in our experiments.

¹⁰ In test prediction, test bag confidences are not defined. This point was also not discussed in Wu et al. [24], so we set all test bag confidences to 1 in our experiments.